(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 288 767 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
05.03.2003 Bulletin 2003/10

(51) Int Cl.[7]: **G06F 1/00**

(21) Application number: **02254593.3**

(22) Date of filing: **28.06.2002**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **04.09.2001 US 944114**

(71) Applicant: **Networks Associates Technology Inc.**
Santa Clara, CA 95054 (US)

(72) Inventors:
- **Barton, Christopher Andrew**
  Buckinghamshire MK18 2RH (GB)
- **Pine, Kristian John**
  Aylesbury, Buckinghamshire HP19 9NL (GB)
- **Gartside, Paul Nichols**
  Milton Keynes MK6 3EP (GB)

(74) Representative:
Robinson, Nigel Alexander Julian et al
D. Young & Co.,
21 New Fetter Lane
London EC4A 1DA (GB)

(54) Updating computer files

(57) A computer file update triggering technique uses tags embedded within e-mail messages sent to connected computers to indicate the existence of an updated version of a computer file to those connected computers. The connected computers may then automatically download the updated version of the computer file. The notification via e-mail of the existence of the updated computer files may be provided as a subscription service by the computer file provider.

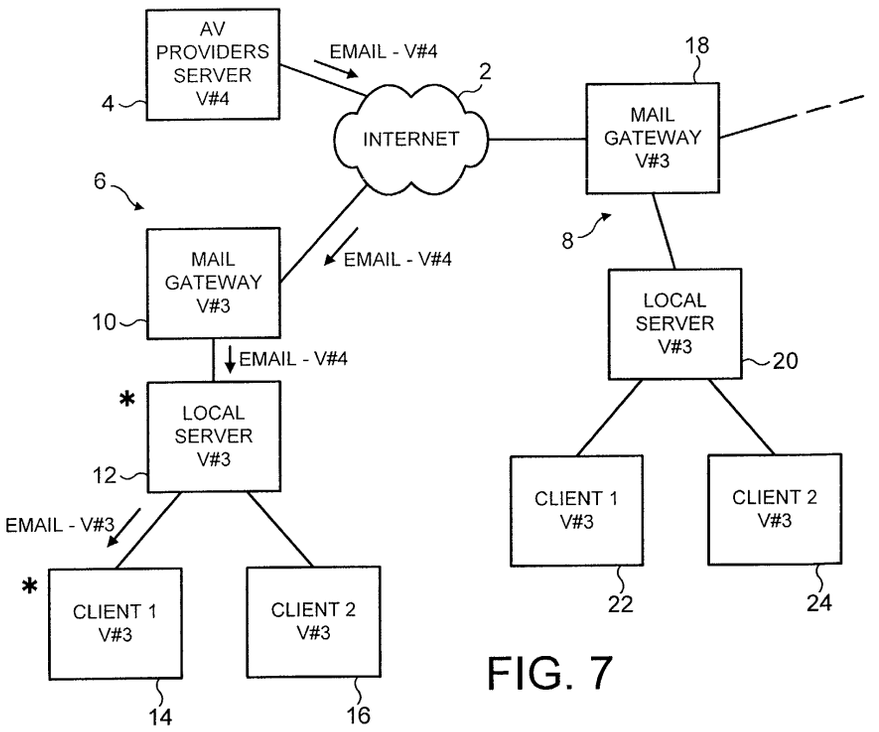

FIG. 7

EP 1 288 767 A2

Description

[0001] This invention relates to the field of computers. More particularly, this invention relates to mechanisms for the updating of computer files

[0002] A problem in the field of computing is the requirement for regular updating of computer files (possibly by downloading a complete new version of the file or by an incremental update in which modifying data for modifying the existing file to form the updated version is downloaded) held on many different computers. A software update may be required because the program has altered in response to the occurrence of bugs within the program or to add additional functionality to a program. Another need for frequent computer file updates is when the computer file represents rapidly evolving data needed by the computer. An example of this is the computer virus definitions data that is used by many anti-virus computer programs. This computer virus definition data is typically updated when a new virus is encountered such that the anti-virus software may provide countermeasures to the new virus. In order that the anti-virus software being used may operate in an effective manner it is important that it should use the most up to date virus definition data.

[0003] In response to this need, anti-virus software suppliers often provide download facilities from which users can download the most up to date versions of the computer virus definition data. One problem with this approach is that a user must know that an updated virus definition data file is present in order that it should be downloaded. One way to deal with this is to configure the computer program software to automatically check for new computer virus definition data at periodic intervals. If these intervals are made too short, then this presents an unnecessary burden upon the computer systems involved. Conversely, if the intervals are made too large, then a significant update required to deal with a new virus threat may not be downloaded in sufficient time to adequately protect from that virus threat.

[0004] A further problem associated with the downloading of virus definition data from the anti-virus software supplier is that peak demand for the download of the new data may cause the systems to malfunction. Computer viruses are becoming increasingly common and destructive. With this background, the release of a new computer virus attracts considerable media attention resulting in many users simultaneously trying to download the updated data in a manner that causes this process to fail.

[0005] Various techniques for updating software are disclosed in US-A-5,940,074, US-A-4,763,271, US-A-5,919,247, US-A-5,577,244, US-A-5,809,287, US-A-5,933,647 and US-A-5,732,275. A technique for updating anti-virus DAT files via a "push" method is disclosed in US-A-6,035,423.

[0006] With conventional "pull" techniques for updating computer files the provider of the updated computer file has relatively little control over the download demand. In critical situations the resources of the provider to enable download by remote computers may be overwhelmed resulting in denial of download to some users. It may be that in these circumstances download of an updated computer file to highly critical corporate mail servers, firewalls and file servers would be denied whereas download to a relatively unexposed home user would be granted.

[0007] Viewed from one aspect this invention provides a method performed by a provider of a computer file to trigger updating of said computer file used by a computer, said method comprising the steps of:

(i) providing an updated version of said computer file at a location from which it may be downloaded by said computer;
(ii) sending a tag indicative of availability of said updated computer file to said computer.

[0008] The invention provides a mechanism whereby the provider of a computer file that is updated may trigger the pull downloading (or other update mechanism) of that updated computer file by their customers in a manner which gives the provider control over which computers are triggered to download the computer file. In particular, known highly critical computers and/or computers at a high risk may be immediately triggered to download the updated computer file as soon as it is available, and potentially automatically without requiring administrative intervention, whilst lower priority computers or computers at less risk may not be so triggered. Furthermore, the provision of such a triggering mechanism to automatically initiate a high priority download of the latest version of a computer file may be provided as a service to specific customers, possibly associated with a subscription fee.

[0009] It will be appreciated that the tag to initiate the downloading could take a wide variety of different forms. As an example, it would be possible for the tag to be embedded within internet data being passed to a computer via the provider's proxy server, firewall, gateway etc. However, in preferred embodiments of the invention the tag forms part of an e-mail message, such as being embedded within the header of an e-mail message, which is sent from the provider to the computer (user) which is registered for the service.

[0010] The widespread availability of e-mail messaging mechanisms and their association with anti-virus scanning systems makes them a particularly convenient way for passing such tag messages and having the tags recognised by appropriate software that will perform the updating.

[0011] It will be appreciated that the computer file to be updated could take a wide variety of forms, such as database data, computer program data etc, but the invention is particularly well suited to the field of anti-virus computer systems in which the computer file to be up-

dated may be computer virus definition data, anti-virus computer programs or scanning engine programs.

[0012] In preferred embodiments the provider of the computer file and the service maintains a database of addresses to be sent the tag message when the updated computer file becomes available. A user may pay to be included within this database and accordingly receive the tag and priority updating. Different levels of service may be provided to different users depending upon the nature of a user, and possibly the amount of the fee paid, with priority data being associated within the database with the data identifying each computer and this priority data being used to control when the tag is sent to the computer concerned.

[0013] The computer to be updated and the storage location of the updated computer file could be physically separated and connected via a remote link, such as an internet link. In some circumstances the source and the computer to be updated may be more locally provided. It will be appreciated that the source of the tag sent to the computers may be physically separated from the storage location of the updated computer file. Furthermore, a plurality of storage locations of the updated computer file could be provided for use by different computers and one or more sources of tag sent to computers may similarly be provided in the same or different locations.

[0014] Complementary aspects of the present invention also provide a computer program product for controlling a computer in accordance with the above-described technique and an apparatus for performing the above-described technique.

[0015] Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figures 1, 2 and 3 illustrate operation of the technique of the present invention in distributing notification of the existence of an update and triggering download of an update;

Figure 4 is a flow diagram showing the processing performed by a peer computer of the type illustrated in Figures 1 to 3;

Figure 5 schematically illustrates a tag of the type that may be inserted within an e-mail message header;

Figure 6 illustrates a computer apparatus that may be used to implement the technique of the present invention;

Figure 7 schematically illustrates the triggering of an update by sending an e-mail message from the update provider;

Figure 8 schematically illustrates subsequent updating operation following the update of Figure 7; and

Figure 9 is a flow diagram schematically illustrating the processing that may be performed by the provider's computer.

[0016] Figure 1 shows a plurality of computers linked via the internet 2 (or some other communication link). A software supplier provides an FTP server 4 from which updated versions of a computer file may be downloaded. Figure 1 illustrates a first local area network 6 and a second local area network 8 linked to the internet 2. The first local area network 6 includes a mail gateway computer 10, a local server 12 and two client workstations 14, 16. In a similar way the second local area network 8 includes a mail gateway 18, a local server 20 and two client workstations 22, 24.

[0017] Using known e-mail protocols and computer programs, the various workstation computers 14, 16, 22, 24 shown in Figure 1 exchange e-mail messages. In alternative embodiments the data exchange could take other forms, such as internet web pages or word processing files that contain headers or the like in which tags for triggering updates may be embedded. All of the computers illustrated in Figure 1 utilise anti-virus software that requires access to an up to date computer virus definition data file. As users of this computer virus definition data file the various mail gateway, server and workstation computers are peers. The illustrated starting situation in Figure 1 is that all of the computers are running the most up to date version (#3) of the computer file that is the version currently stored on the FTP server 4.

[0018] Figure 2 illustrates the start of the dissemination of an update. The software supplier loads an updated version of the computer virus definition data file onto the FTP server 4. A manual or an automatic timed update of the mail gateway 10 then takes place that updates the version of the computer file on the mail gateway to #4. Subsequently, an e-mail message is issued by the workstation computer 16 destined for the workstation computer 24. The anti-virus software within the workstation computer 16 tags the e-mail header of the e-mail message with the version #3 of the computer virus definition data file that the workstation computer 16 is currently using. This e-mail message first passes through the local server 12 which is also using this same version #3 and so leaves the tag unaltered. When the e-mail message reaches the mail gateway 10, the mail gateway checks the tag within the e-mail message header and determines that it is itself using a more up-to-date version of the computer file in question and so replaces the tag with one that indicates that version #4 has been used and is available. The tag may also indicate whether or not that computer may itself serve as the source of the update data for other computers. This e-mail message then propagates via the internet 2 and

through the mail gateway 18 and the local server 20 until it reaches the workstation computer 24. Each of the mail gateway 18, the local server 20 and the workstation computer 24 examines the tag within the e-mail message header and determines that it indicates the existence of a more up to date version of the computer virus definition data file than that which they are currently using. Accordingly, each of the mail gateway 18, the local server 20 and the workstation computer 24 is triggered (after an initial delay period) to download the updated version of the computer file from the FTP server 4 (a predetermined source of updates). The "*" indicates that a particular peer computer has detected that it should download an updated version of the computer file. The mail gateway 18, the local server 20 and the workstation computer 24 will continue to use version #3 until they have the updated version #4. The update mechanism used is preferably the pre-existing standard "pull" mechanism, but it is envisaged that alternative emergency or special purpose update mechanisms for use when triggered by a received tag could be provided.

[0019] Figure 3 shows the situation when the updates to the mail gateway 18, the local server 20 and the workstation computer 24 have all taken place. In the example shown, the user of the workstation computer 24 sends an e-mail message to the user of the workstation computer 16 that is also copied to the user of the workstation computer 22. As the workstation computer 24 that is the source of this e-mail message has now been updated to version #4, it includes within the header of the e-mail message a tag indicating that version #4 exists. When the workstation computer 22 receives this message, this is detected as indicating that the workstation computer 22 should download the updated version #4 from the FTP server 4. The e-mail message also propagates via the local server 20, the mail gateway 18 and the mail gateway 10 towards the other target recipient that is the workstation computer 16. All of these peer computers have already been updated, thus they do not action version #4 tags and below. The first computer reached in this transmission path that has not yet been updated is the local server 12 and the second is the workstation computer 16. Both of these computers also detect that the tag shows a version level #4 higher than that which they are currently using #3 and accordingly trigger the download of an update from the FTP server 4.

[0020] It will be appreciated that the mechanisms shown in Figures 1, 2 and 3 allow for the automatic and rapid dissemination of the information that an updated file exists. Furthermore, those computers that receive more e-mail messages are more likely to receive this notification sooner. These are the very computers that are generally at a high risk from computer viruses and accordingly it is appropriate that they should be the first that download the updated version of the computer virus definition data file. A little used computer will only download its update at a later time, and yet this will pose potentially lower level of risk since the little used computer is less likely to receive an infectious element.

[0021] Figure 4 is a flow diagram schematically illustrating the processing performed by a particular peer computer.

[0022] At step 26, the computer receives an e-mail message. At step 28 the computer searches the message header of the e-mail for any header tag present and decrypts this if it is found. In some embodiments the header tag may not be encrypted.

[0023] At step 30, a test is made as to whether any header tag has been found. If a header tag has not been found, then the e-mail message is scanned for computer viruses using the existing anti-virus software at its current level of update and a header tag added to the e-mail message indicative of that current level of update at step 32.

[0024] If a header tag is found at step 30, then step 34 tests whether or not that header tag indicates a version of the software that is older than that held by the computer performing the process illustrated in Figure 4. If the received header tag is indicative of an older version, the processing proceeds to step 32 at which the e-mail message is scanned using what is known to be more up to date data than has previously been applied to that message and the header tag indicative of that more up to date data is added to the e-mail header. The existing tag indicative of the older version of the computer file may or may not be removed. The tag may also include parameters indicative of previous processing applied to the message, such as the program options set (e.g. all files, macro heuristics, all heuristics) on previous anti-virus scans applied to the message. These parameters can be used to determine whether or not further scanning is to be applied by the computer currently processing the message.

[0025] If the test at step 34 indicates that the received e-mail message included a header tag that was not older than the local version, then processing proceeds to step 36. Step 36 tests whether or not the tag of the received e-mail message indicates a newer version of the computer file is available. If a newer version is not available, then processing proceeds to step 37. Step 37 decided whether or not the message should be scanned at step 32 in dependence upon parameters set on the processing computer and parameters within the tag as mentioned above that indicate in more detail what previous scanning has been applied to the message. If the test at step 36 indicates that a newer version of the computer file than that stored by the local computer is available, then processing proceeds to step 40.

[0026] Step 40 tests how may versions ahead of the current version the tag within the received e-mail message indicates is available. If this number exceeds a predetermined threshold N, then this is indicative of some malfunction or malicious interference with the message tags and accordingly processing proceeds to step 32.

[0027] If the test at step 40 indicates that the updated version is less than the threshold number N ahead of

the currently used version, then processing proceeds to step 42 which scans using the currently held version and then imposes an initial delay before processing proceeds to step 44 where an update attempt is triggered to download the updated version of the computer file from a remote source. The remote source may be an FTP server 4 linked via the internet 2 (or any other link) to the computer in question, or could be a server file location within a local area network 6, 8 or some other source.

[0028] At step 46, a test is made as to whether or not the update attempt has failed. If the update attempt has not failed, then processing continues with other normal e-mail processing operations at step 38.

[0029] If the update attempt has failed, then processing proceeds to step 48 which imposes a psuedo-random failure delay prior to returning processing to step 44 to attempt another update. A predetermined number of update attempt failures may trigger a user warning message.

[0030] Figure 5 illustrates a message tag of the type that may be inserted within an E-mail message header. The "X-" prefix in the tag is one defined some in standard e-mail protocols as indicating that the information that follows on the line is for information purposes only and is not actively processed in normal systems. In the illustrated example, the tag starts with a coding for "McAfee-Sig:". This is a code sequence that is searched for by peer computers within e-mail message headers to detect the presence of a tag indicating what version levels of an anti-virus system have already been applied to that e-mail message. This version information follows in the form of "<S#-xxxE#-yyyD#-zzz>", portions of which respectively indicate the anti-virus software program version number, the computer virus detection engine version number and the computer virus definition data version number. In practice, this version information may be encrypted to make it more difficult to tamper with the information in an attempt to misdirect or interfere with the update mechanisms. Various known encryption techniques may be used. The tag could also include parameters indicating previously applied scan options or other data and could extend over more than one line.

[0031] Figure 6 schematically illustrates a computer 50 of the type that may be used to implement the techniques described above (more simple appliance type devices could also be used). The computer 50 includes a central processing unit 52, a read only memory 54, a random access memory 56, a network link 58, a hard disk drive 60, a display driver 62, a display 64, a user input/output driver 64, a keyboard 66 and a mouse 68.

[0032] In operation, the central processing unit 52 executes computer programs stored upon the hard disk drive 60 or within the read only memory 54 using the random access memory as working memory. User inputs for controlling the computer 50 are received from the keyboard 66 and the mouse 68 via the user input/output unit 64. Processing results may be displayed to the user using the display 64 via the display driver unit 62.

[0033] In operation, an e-mail message may be received from the internet 2 via the network link 58 into an e-mail program being executed by the central processing unit 52. Part of this e-mail program may trigger an anti-virus scan of the received e-mail. This anti-virus scan includes the processing illustrated in Figure 4 and accordingly detects if the received e-mail message includes the information that an updated version of any of the components of the anti-virus system exists for download. Should such updated versions exist, then they may be downloaded by the computer 50 under program control from a remote source, such as an FTP server 4, via the internet 2 and the network link 58.

[0034] The updated computer files, such as the anti-virus software program, the search engine program or the virus definitions, will typically then be stored on the hard disk drive 60 of the computer 50. The program that causes the processing of Figure 4 to take place will also typically be stored upon the hard disk drive 60. The computer program may be distributed via a recordable medium, such as a floppy disk or a CD, or may itself be downloaded via the network link 58.

[0035] The computer 50 may pass the e-mail message onto another computer or may itself originate a new e-mail message. In either case, any outbound e-mail message is marked with a tag indicating the version levels of the anti-virus software components used by the computer 50 if these are more up to date than any indications already within the e-mail message.

[0036] Figure 7 schematically illustrates a system in which a system sends an e-mail message to a client workstation 14 to trigger updating of a computer file made available on the FTP server 4. As an example, a computer virus definition data file may be updated to include data defining a newly released computer virus. The computer virus definition data provider will then place this updated computer file on the FTP server 4, that server not necessarily using that computer file itself, but rather providing a storage location from which remote computers may download that computer file via the internet 2.

[0037] The FTP server 4, or possibly a different computer, will then read a database of e-mail addresses to which an e-mail is to be sent including a tag indicating that the new version #4 of the computer file is available for download. In this example, the e-mail message is sent to the client workstation 14 and passes to this client workstation 14 via the mail gateway 10 and the local server 12, all of which are using the previous version of this computer file, namely version #3. Accordingly, in accordance with the previously described techniques each of the mail gateway computer 10, the local server 12 and the client workstation computer 14 examines the tag within the e-mail header and notes that a more up to date version of the computer file in question is available for download and then triggers the update process (the

update process could take a variety of different forms, such as local updates etc, and may incorporate load balancing provisions such as retrying a download after a random delay if a connection is refused). For example, in this way, the highly critical gateway 10 may be triggered by the provider themselves to start a pull download of the updated computer file as soon as it is available. The owner of the local area network 6 may pay a subscription to be so notified by the provider.

[0038] Figure 8 illustrates subsequent operation in a manner similar to Figure 2. In Figure 8 the client workstation computer 16 sends an e-mail message to the client workstation computer 22. Since the first local area network 6 has had the local server 12 and the mail gateway 10 updated to the latest version of the computer file, i.e. version #4, as the e-mail message propagates along its path the tag within the header is updated to indicate that this latest version is available and accordingly the mail gateway computer 18, the local server 20 and the client workstation computer 22 all read this tag within the header and note that a more up to date version of the computer file is available, thus, these further computers may then initiate a pull download of the updated computer file from the FTP server 4. As an alternative, the provider may arrange that the software on the mail gateway 10 does not pass out the updated tag thereby preventing redistribution of the subscription service.

[0039] Figure 9 schematically illustrates the processing that may be performed by the computer of the computer program provider. At step 70 the updated computer file is placed on the provider's server to be available for download via the internet. At step 72 the provider accesses their database of e-mail addresses to be sent triggering e-mails. The recipients of these e-mails may pay a subscription to be included within this database. The database may include a priority ordering which will control the timing at which the triggering e-mails are sent out from the provider, thus, the highest priority customers may be sent their triggering e-mails immediately, with lower priority customers being sent their e-mails after a predetermined delay, such as an hour, in order to allow the highest priority customers to obtain their updated versions of the computer files whilst the FTP server 4 is relatively lightly loaded. As previously mentioned, if a client computer is unable to connect to download the computer file, it may be arranged to automatically retry after a randomly chosen time delay.

[0040] Step 74 corresponds to the sending of the e-mails containing the tags relating to the availability of the new version of the computer file out to the customers as indicated by the e-mail addresses within the database. When this notification process is complete, then the process of Figure 9 may terminate.

Claims

1. A method performed by a provider of a computer file to trigger updating of said computer file used by a computer, said method comprising the steps of:

(i) providing an updated version of said computer file at a location from which it may be downloaded by said computer;
(ii) sending a tag indicative of availability of said updated computer file to said computer.

2. A method as claimed in claim 1, wherein said tag is part of an e-mail message.

3. A method as claimed in claim 2, wherein said tag is part of an e-mail message header.

4. A method as claimed in any one of claims 1 to 3, wherein said computer is connected to said location via an internet link.

5. A method as claimed in any one of claims 1 to 4, wherein computer file is computer virus definition data.

6. A method as claimed in any one of claims 1 to 5, wherein said computer file is an anti-virus computer program file.

7. A method as claimed in any one of claims 1 to 6, wherein said tag includes data indicative of a version level of said computer file.

8. A method as claimed in any one of claims 1 to 7, comprising maintaining a database of computers to which said tag is to be sent when an updated version of said computer file is made available.

9. A method as claimed in claim 8, wherein said database includes priority data indicating a priority level associated with an address, said priority level being used to control how rapidly after said updated version of said computer file is made available said tag is sent to said computer.

10. A method as claimed in any one of the preceding claims, wherein sending of said tag upon availability of an updated version of said computer file is provided as a subscription service by said provider.

11. A computer program product for use by a provider of a computer file to trigger updating of said computer file used by a computer said provider having provided an updated version of said computer file at a location from which it may be downloaded by said computer, said computer program product comprising:

(i) tag sending code operable to send a tag indicative of availability of said updated computer

file to said computer.

12. A computer program product as claimed in claim 11, wherein said tag is part of an e-mail message.

13. A computer program product as claimed in claim 12, wherein said tag is part of an e-mail message header.

14. A computer program product as claimed in any one of claims 11 to 13, wherein said computer is connected to said location via an internet link.

15. A computer program product as claimed in any one of claims 11 to 14, wherein computer file is computer virus definition data.

16. A computer program product as claimed in any one of claims 11 to 15, wherein said computer file is an anti-virus computer program file.

17. A computer program product as claimed in any one of claims 11 to 16, wherein said tag includes data indicative of a version level of said computer file.

18. A computer program product as claimed in any one of claims 11 to 17, comprising database code operable to maintain a database of computers to which said tag is to be sent when an updated version of said computer file is made available.

19. A computer program product as claimed in claim 18, wherein said database includes priority data indicating a priority level associated with an address, said priority level being used to control how rapidly after said updated version of said computer file is made available said tag is sent to said computer.

20. A computer program product as claimed in any one of claims 11 to 19, wherein sending of said tag upon availability of an updated version of said computer file is provided as a subscription service by said provider.

21. Apparatus for use by a provider of a computer file to trigger updating of said computer file used by a computer said provider having provided an updated version of said computer file at a location from which it may be downloaded by said computer, said apparatus comprising:

(i) a tag sender operable to send a tag indicative of availability of said updated computer file to said computer.

22. Apparatus as claimed in claim 21, wherein said tag is part of an e-mail message.

23. Apparatus as claimed in claim 22, wherein said tag is part of an e-mail message header.

24. Apparatus as claimed in any one of claims 21 to 23, wherein said computer is connected to said location via an internet link.

25. Apparatus as claimed in any one of claims 21 to 24, wherein computer file is computer virus definition data.

26. Apparatus as claimed in any one of claims 21 to 25, wherein said computer file is an anti-virus computer program file.

27. Apparatus as claimed in any one of claims 21 to 26, wherein said tag includes data indicative of a version level of said computer file.

28. Apparatus as claimed in any one of claims 21 to 27, comprising database logic operable to maintain a database of computers to which said tag is to be sent when an updated version of said computer file is made available.

29. Apparatus as claimed in claim 28, wherein said database includes priority data indicating a priority level associated with an address, said priority level being used to control how rapidly after said updated version of said computer file is made available said tag is sent to said computer.

30. Apparatus as claimed in any one of claims 21 to 29, wherein sending of said tag upon availability of an updated version of said computer file is provided as a subscription service by said provider.

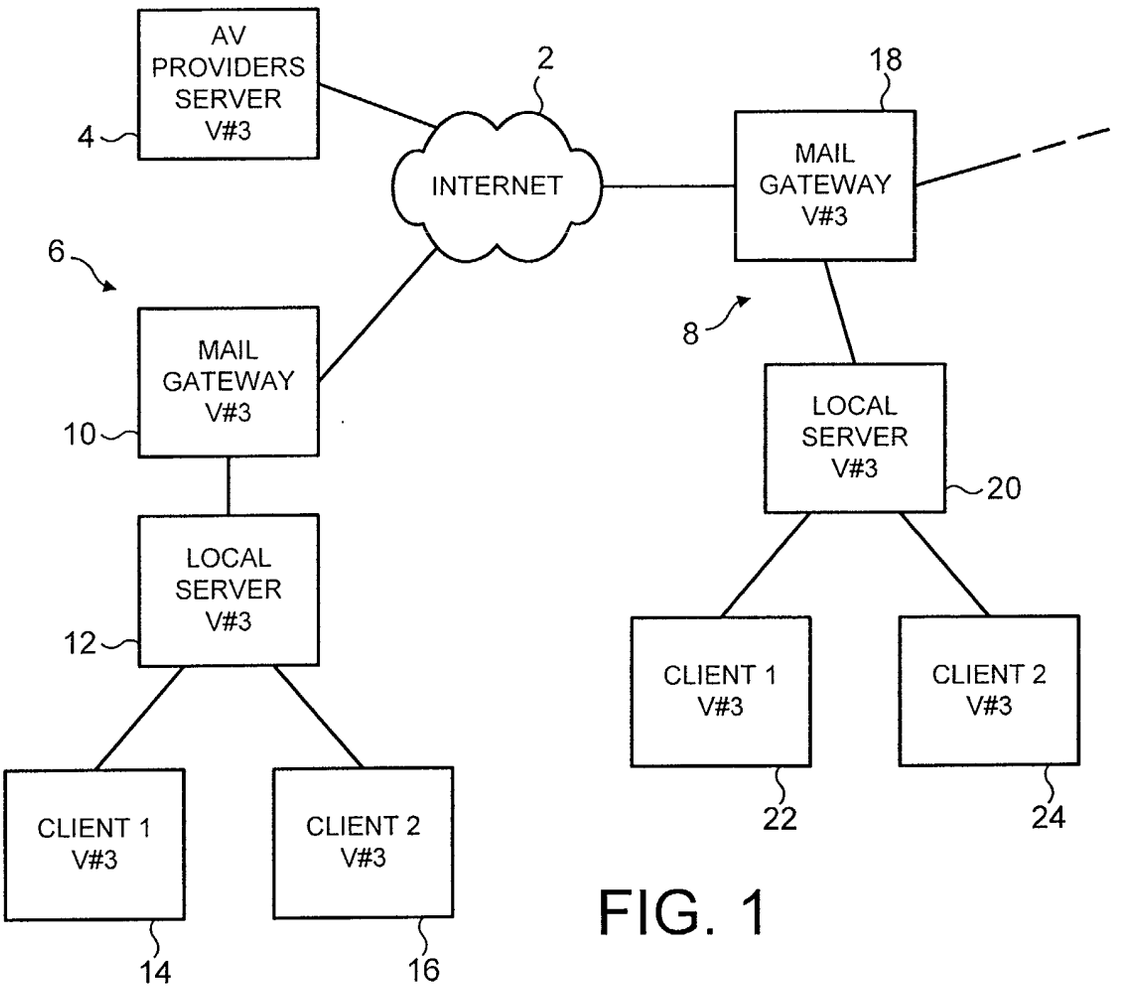
AV PROVIDERS SERVER V#3
4
2
INTERNET
18
MAIL GATEWAY V#3
6
8
MAIL GATEWAY V#3
10
LOCAL SERVER V#3
20
LOCAL SERVER V#3
12
CLIENT 1 V#3
CLIENT 2 V#3
22
24
CLIENT 1 V#3
CLIENT 2 V#3
14
16

FIG. 1

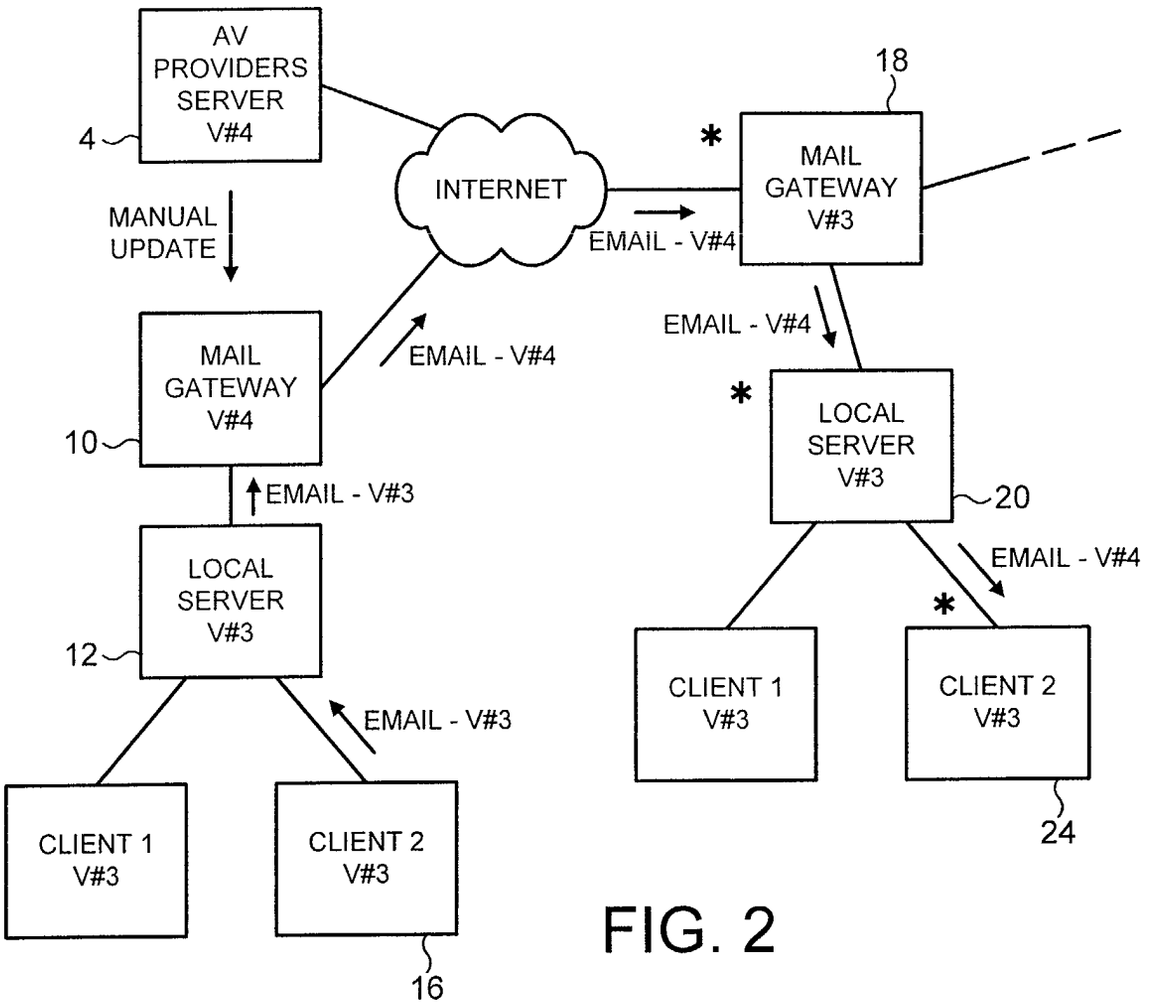
AV PROVIDERS SERVER V#4
4
MANUAL UPDATE
INTERNET
MAIL GATEWAY V#4
10
EMAIL - V#4
EMAIL - V#3
LOCAL SERVER V#3
12
EMAIL - V#3
CLIENT 1 V#3
CLIENT 2 V#3
16
18
MAIL GATEWAY V#3
EMAIL - V#4
EMAIL - V#4
LOCAL SERVER V#3
20
EMAIL - V#4
CLIENT 1 V#3
CLIENT 2 V#3
24

FIG. 2

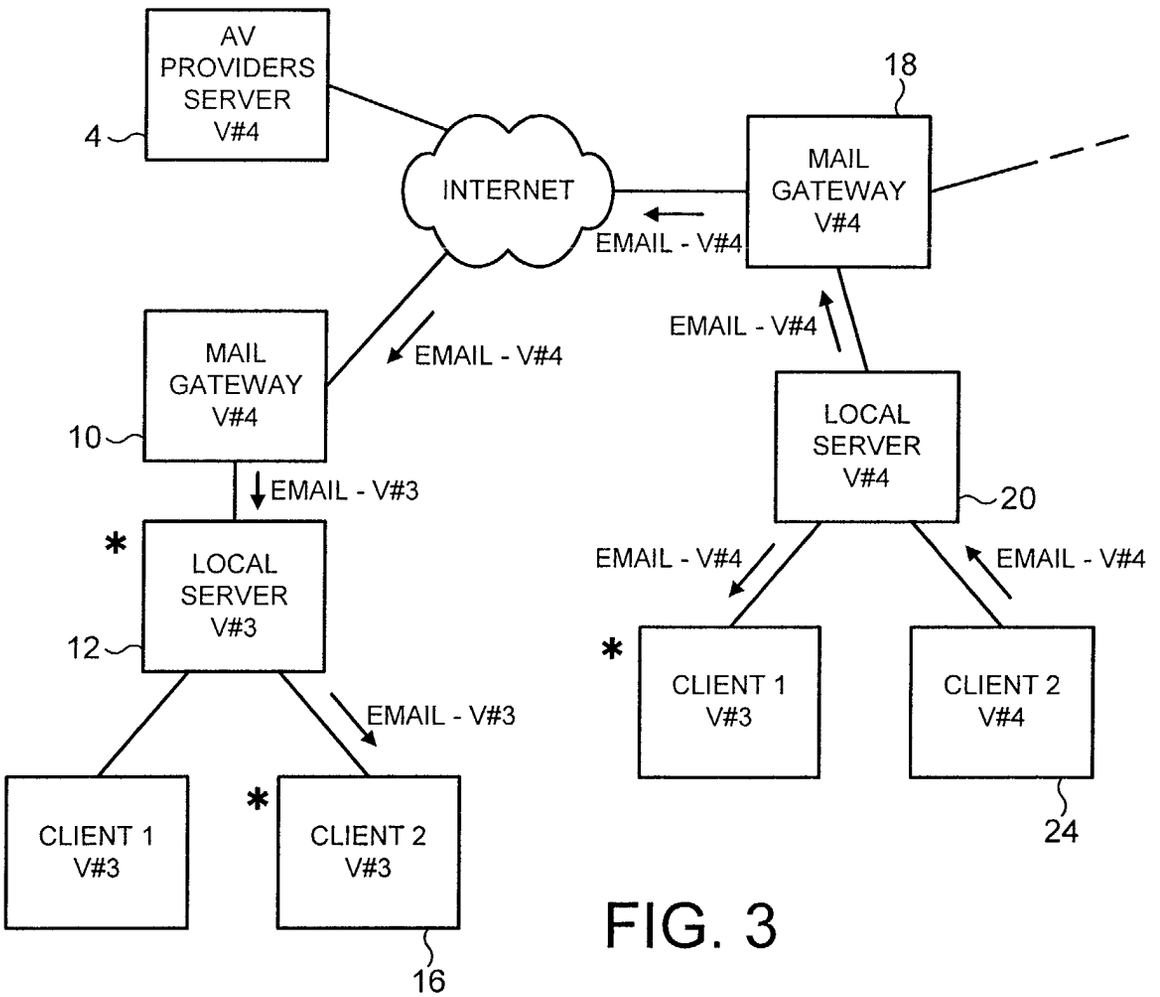
AV PROVIDERS SERVER V#4
4
INTERNET
18
MAIL GATEWAY V#4
EMAIL - V#4
EMAIL - V#4
MAIL GATEWAY V#4
10
EMAIL - V#4
LOCAL SERVER V#4
20
EMAIL - V#3
*
LOCAL SERVER V#3
12
EMAIL - V#4
EMAIL - V#4
*
CLIENT 1 V#3
CLIENT 2 V#4
24
EMAIL - V#3
CLIENT 1 V#3
*
CLIENT 2 V#3
16

FIG. 3

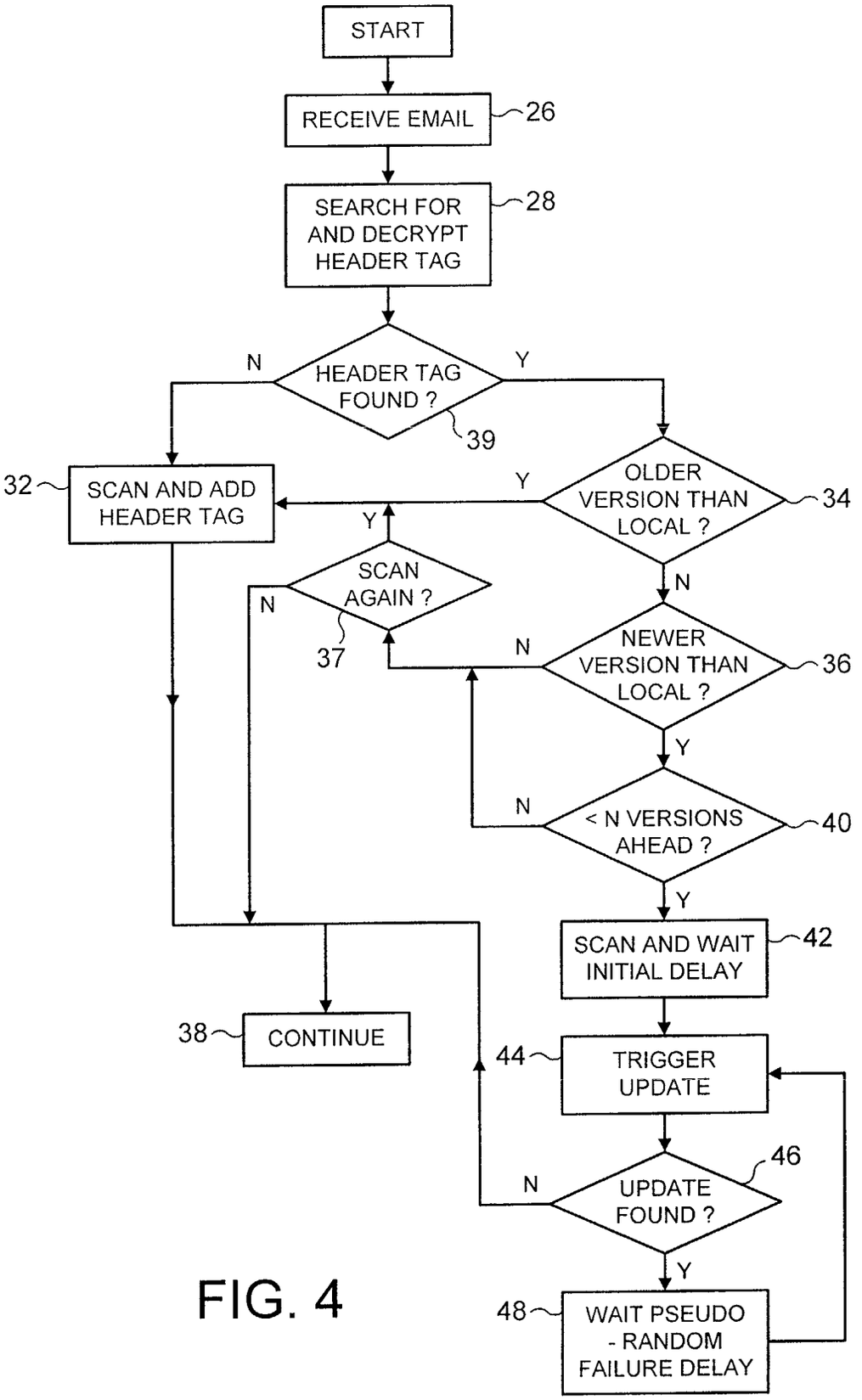
START
RECEIVE EMAIL
26
SEARCH FOR AND DECRYPT HEADER TAG
28
HEADER TAG FOUND ?
39
N
Y
SCAN AND ADD HEADER TAG
32
OLDER VERSION THAN LOCAL ?
34
Y
Y
SCAN AGAIN ?
37
N
N
NEWER VERSION THAN LOCAL ?
36
N
Y
< N VERSIONS AHEAD ?
40
N
Y
SCAN AND WAIT INITIAL DELAY
42
CONTINUE
38
TRIGGER UPDATE
44
UPDATE FOUND ?
46
N
Y
WAIT PSEUDO - RANDOM FAILURE DELAY
48

FIG. 4

X - McAfee-sig: <S#-xxx E#-YYY D#-ZZZ>

FIG. 5

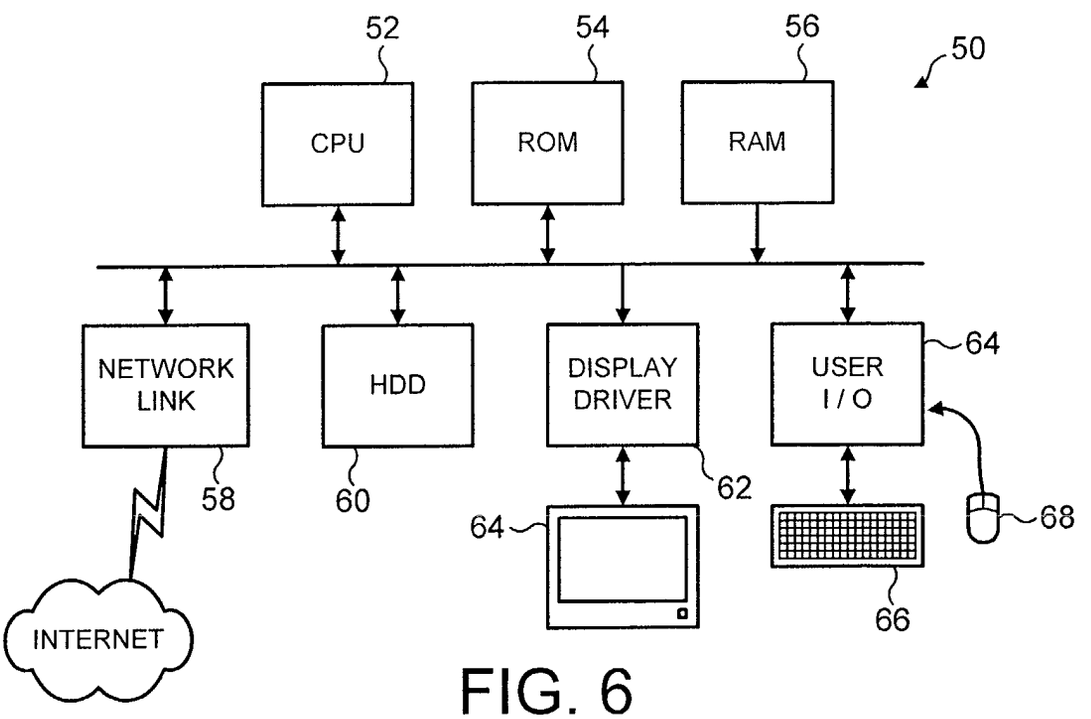

FIG. 6

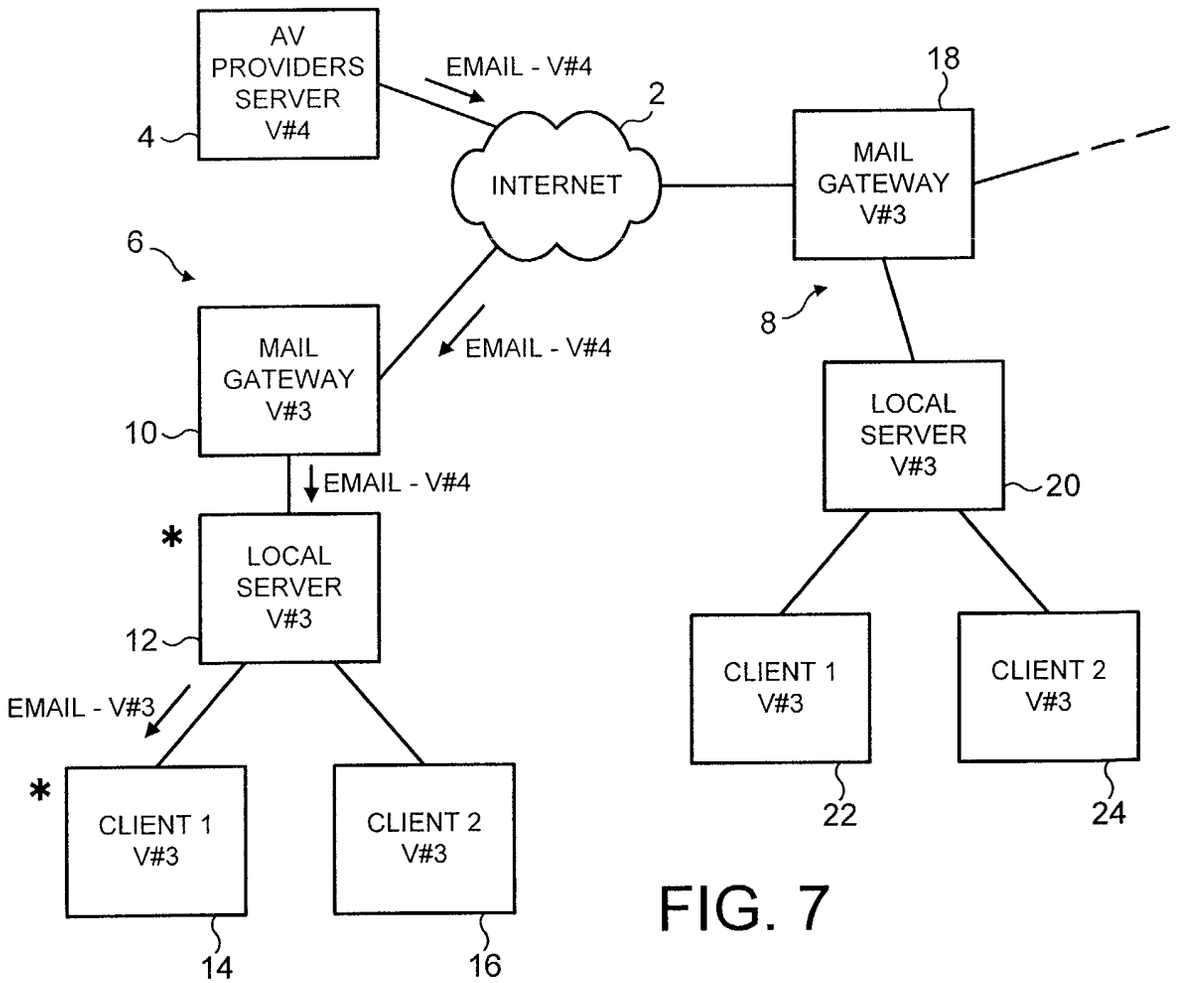
AV PROVIDERS SERVER V#4
4
EMAIL - V#4
2
INTERNET
18
MAIL GATEWAY V#3
6
MAIL GATEWAY V#3
10
EMAIL - V#4
EMAIL - V#4
8
LOCAL SERVER V#3
20
EMAIL - V#4
*
LOCAL SERVER V#3
12
EMAIL - V#3
CLIENT 1 V#3
22
CLIENT 2 V#3
24
*
CLIENT 1 V#3
CLIENT 2 V#3
14
16

FIG. 7

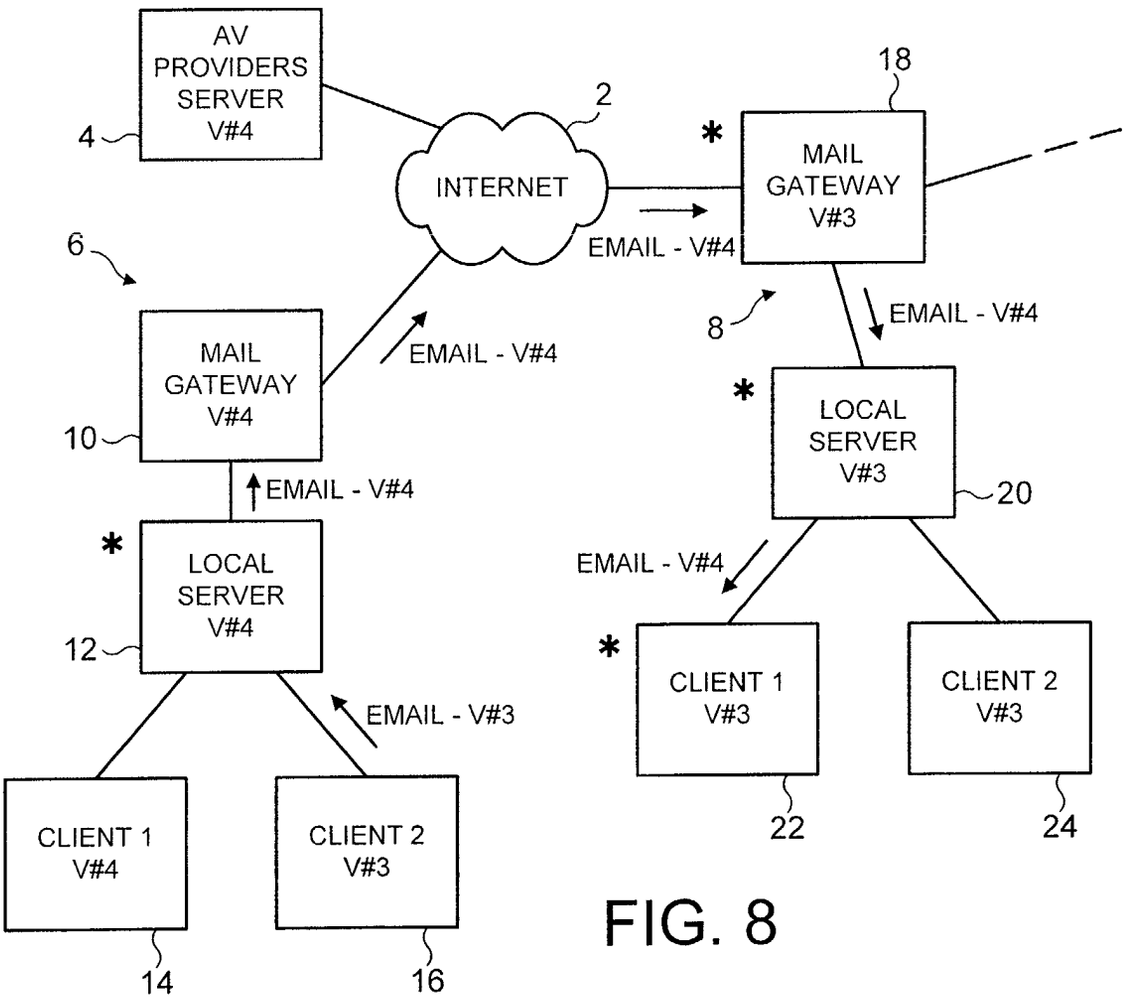
AV PROVIDERS SERVER V#4
4
INTERNET
2
MAIL GATEWAY V#3
18
EMAIL - V#4
6
MAIL GATEWAY V#4
10
EMAIL - V#4
8
EMAIL - V#4
LOCAL SERVER V#3
20
EMAIL - V#4
LOCAL SERVER V#4
12
EMAIL - V#3
CLIENT 1 V#3
22
CLIENT 2 V#3
24
CLIENT 1 V#4
14
CLIENT 2 V#3
16

FIG. 8

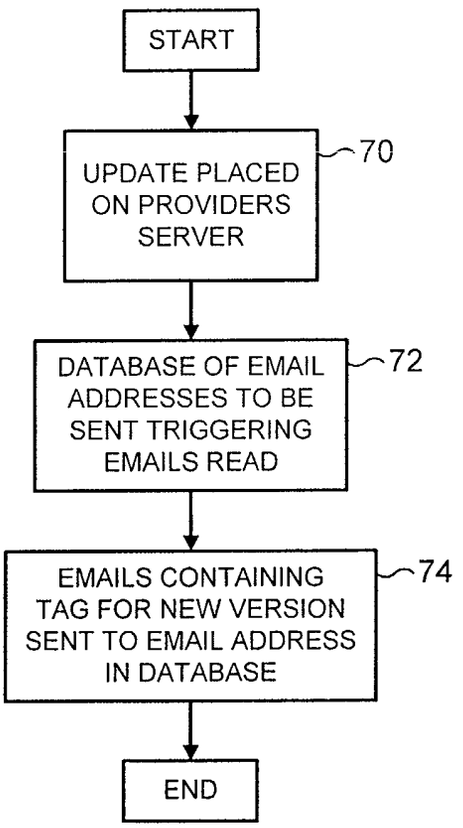
START
UPDATE PLACED ON PROVIDERS SERVER
70
DATABASE OF EMAIL ADDRESSES TO BE SENT TRIGGERING EMAILS READ
72
EMAILS CONTAINING TAG FOR NEW VERSION SENT TO EMAIL ADDRESS IN DATABASE
74
END

FIG. 9